Master Informatics Eng.

2020/21

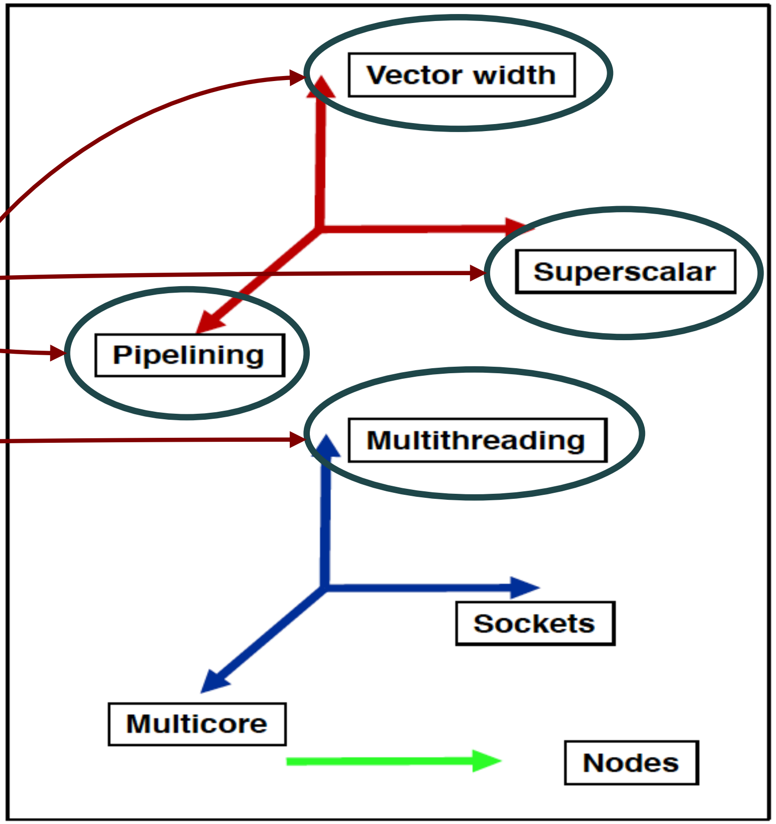
A.J.Proença

From ILP to Multithreading

(most slides are borrowed)

Key issues for parallelism in a single-core

- **Last week & this week:**
  - pipelining:
    reviewed in the `combine` example
  - superscalar:
    idem, but some more this week
  - multithreading:
    hw support for multithreading
- **Next 2 weeks:**
  - memory hierarchy:
    review of multilevel caching
  - vector computing & extensions:
    vector computers & vector extensions to scalar processors



Multiple Issue and Static Scheduling

- Pipelining takes CPI→1
- To achieve CPI < 1, need to complete multiple instructions per clock cycle
- Solutions:
  - statically scheduled superscalar processors
  - VLIW (very long instruction word) processors
  - dynamically scheduled superscalar processors

Multiple Issue

Common name	Issue structure	Hazard detection	Scheduling	Distinguishing characteristic	Examples
Superscalar (static)	Dynamic	Hardware	Static	In-order execution	Mostly in the embedded space: MIPS and ARM, including the ARM Cortex A8 , Atom
Superscalar (dynamic)	Dynamic	Hardware	Dynamic	Some out-of-order execution, but no speculation	None at the present
Superscalar (speculative)	Dynamic	Hardware	Dynamic with speculation	Out-of-order execution with speculation	Intel Core i3, i5, i7; AMD Phenom; IBM Power 7
VLIW/LIW	Static	Primarily software	Static	All hazards determined and indicated by compiler (often implicitly)	Most examples are in signal processing, such as the TI C6x
EPIC	Primarily static	Primarily software	Mostly static	All hazards determined and indicated explicitly by the compiler	Itanium

EPIC: Explicitly Parallel Instruction Computer

Speculative execution

From Wikipedia, the free encyclopedia

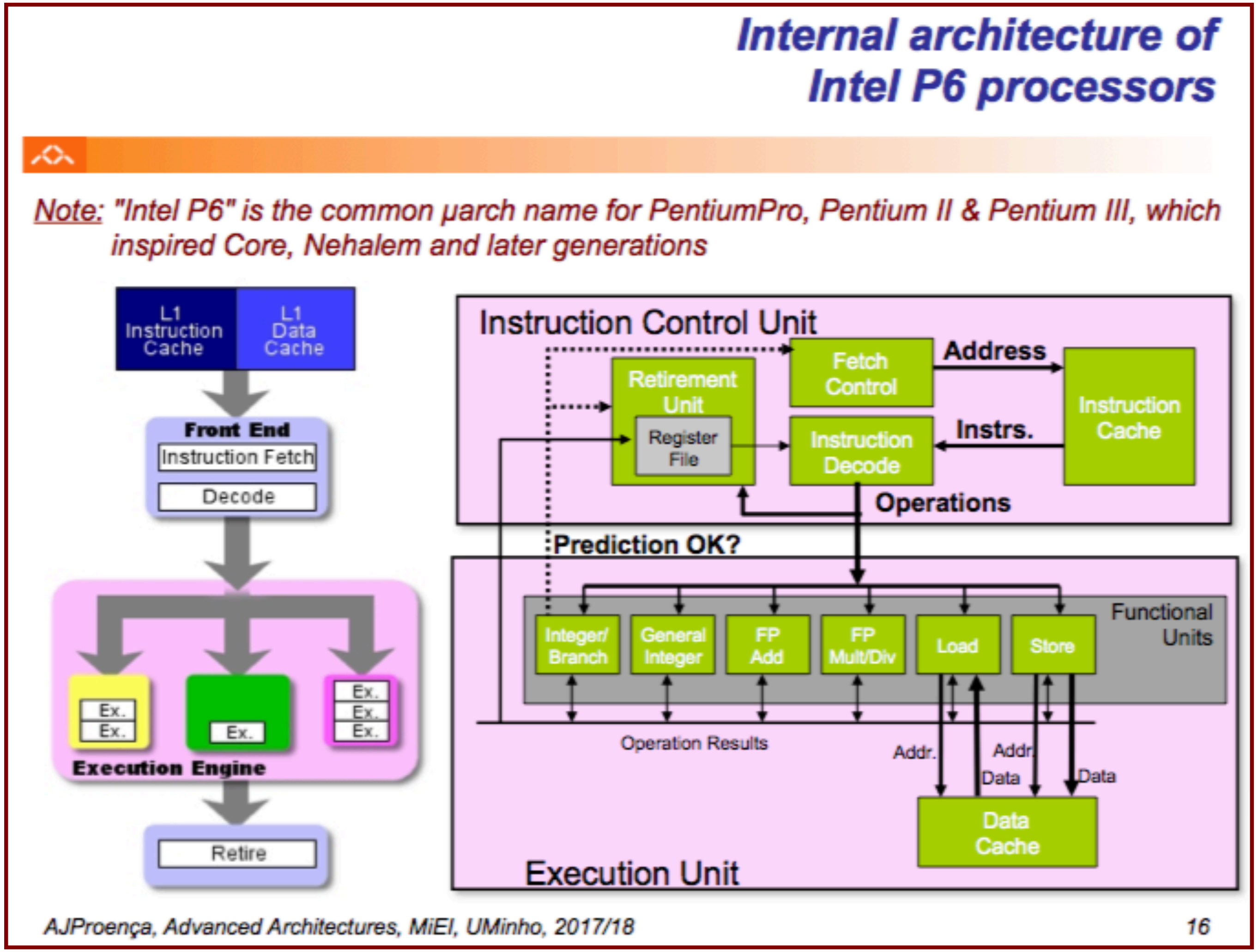
Speculative execution is an optimization technique where a computer system performs some task that may not be needed. Work is done before it is known whether it is actually needed, so as to prevent a delay that would have to be incurred by doing the work after it is known that it is needed. If it turns out the work was not needed after all, most changes made by the work are reverted and the results are ignored.

The objective is to provide more concurrency if extra resources are available. This approach is employed in a variety of areas, including branch prediction in pipelined processors, value prediction for exploiting value locality,[1] prefetching memory and files, and optimistic concurrency control in database systems.[2][3][4]

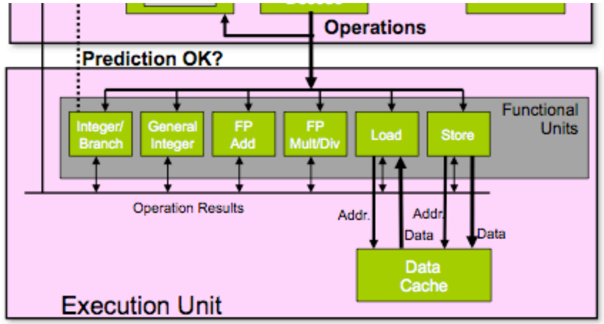
Speculative multithreading is a special case of speculative execution.

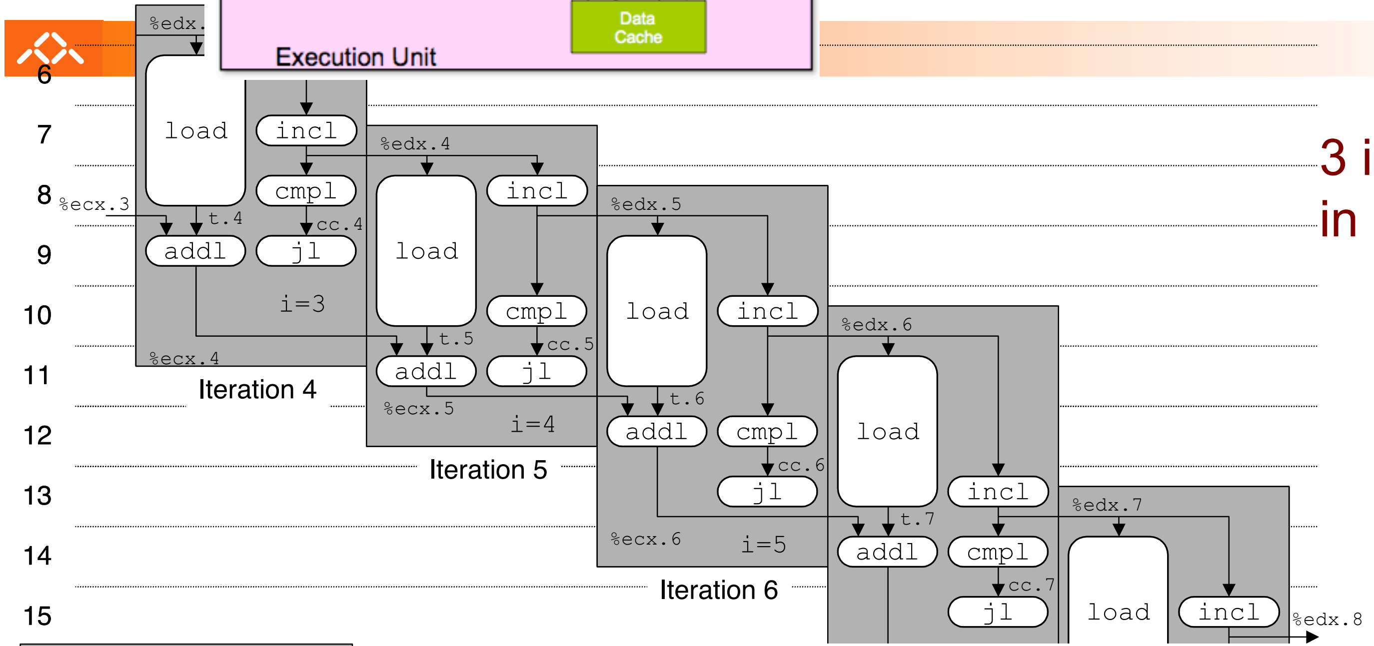
The n-way superscalar P6: how this architecture supports n-way multiple issue

As seen before...



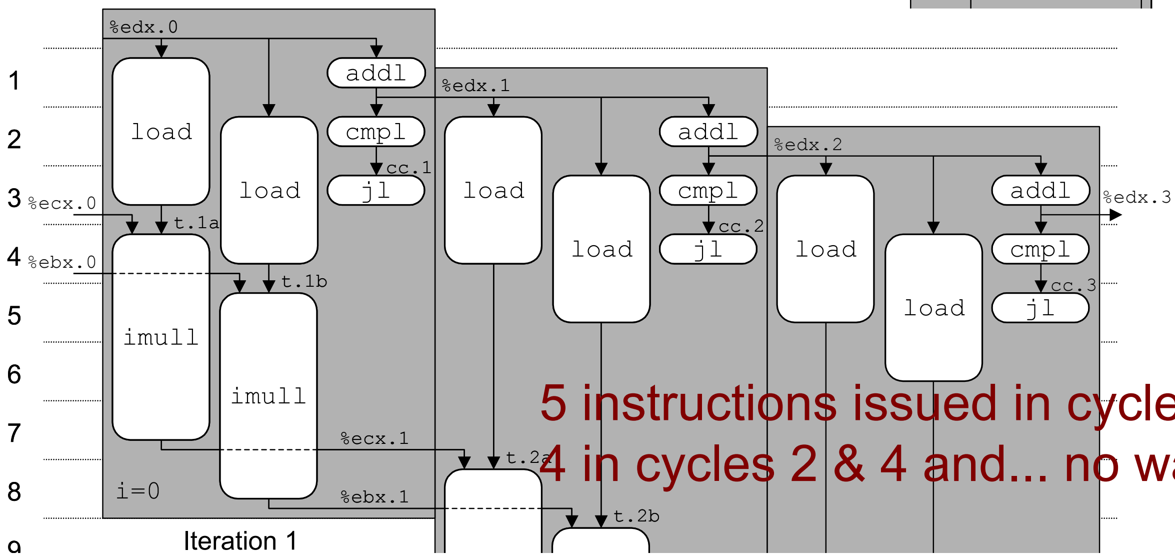
Executions views of `combine`: can this architecture be 3-way?



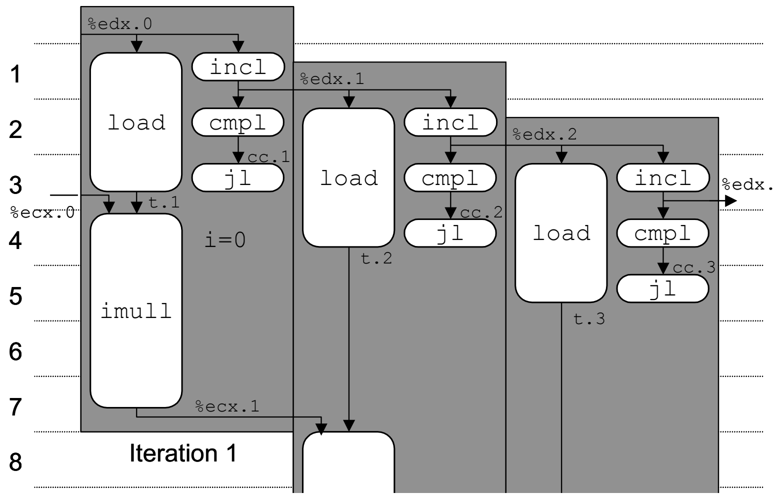


3 instructions issued in cycles 8, 9... ok!

4 instructions issued in cycle 3... no way!



5 instructions issued in cycle 3, 4 in cycles 2 & 4 and... no way!



Multithreading

Performing multiple threads of execution in parallel

- Replicate registers, PC/IP, etc.
- Fast switching between threads

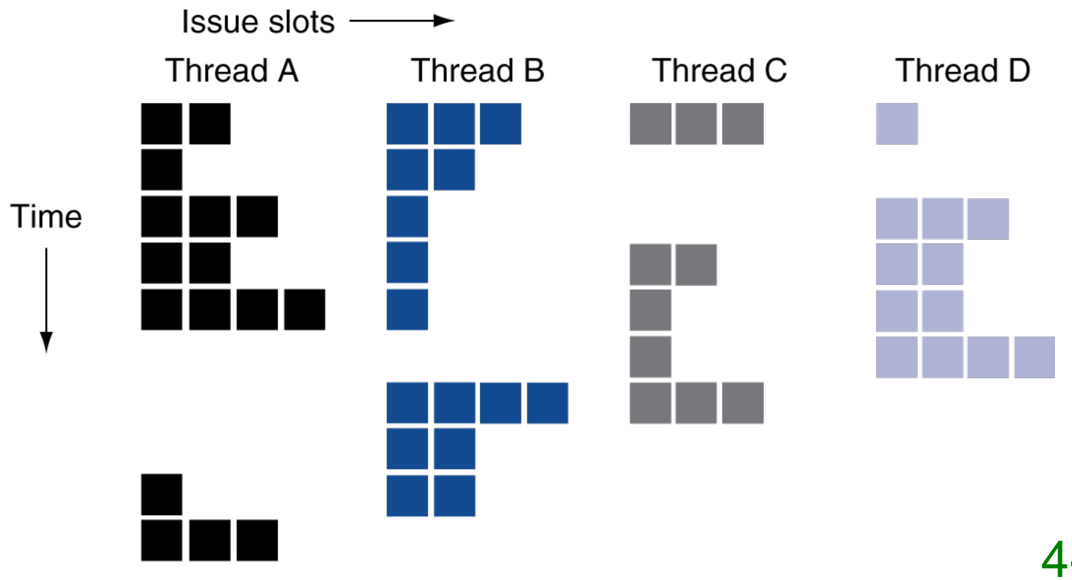
1. Fine-grain multithreading / time-multiplexed MT
   - Switch threads after each cycle
   - Interleave instruction execution
   - If one thread stalls, others are executed
2. Coarse-grain multithreading
   - Only switch on long stall (e.g., L3-cache miss)
   - Simplifies hardware, but doesn't hide short stalls (eg, data hazards)

3. Simultaneous Multithreading

- In multiple-issue dynamically scheduled processor
  - Schedule instructions from multiple threads
  - Instructions from independent threads execute when function units are available
  - Within threads, dependencies handled by scheduling and register renaming
- Example: Intel from Pentium-4 HT
  - Two threads: duplicated registers, <u>shared</u> function units and caches

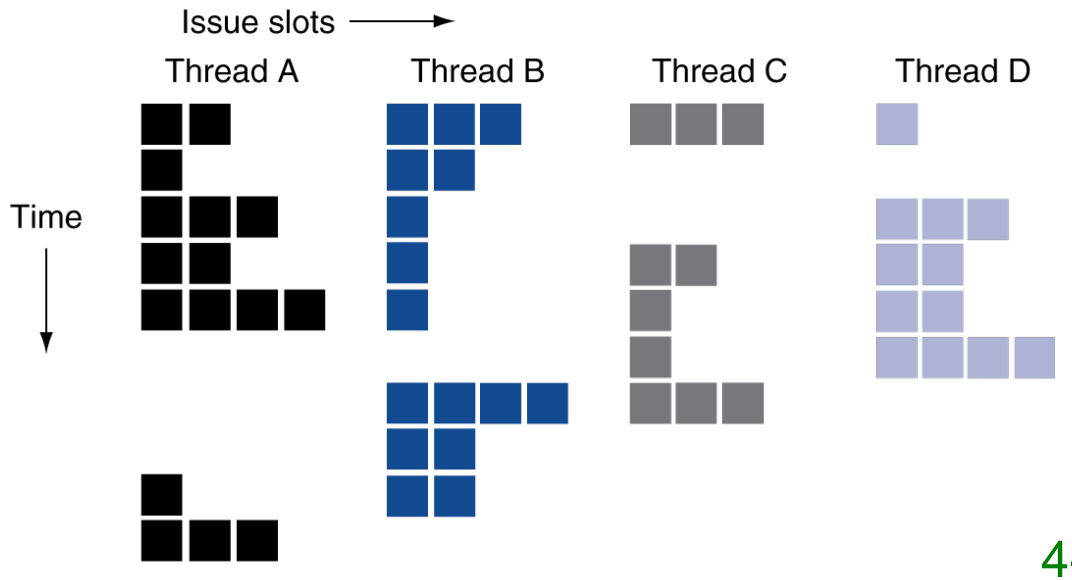
***HT**: Hyper-Threading, Intel trade mark for their SMT implementation*
*MT in Xeon Phi KNC: 4-way SMT with time-mux MT, **not HT**!*

Multithreading Example

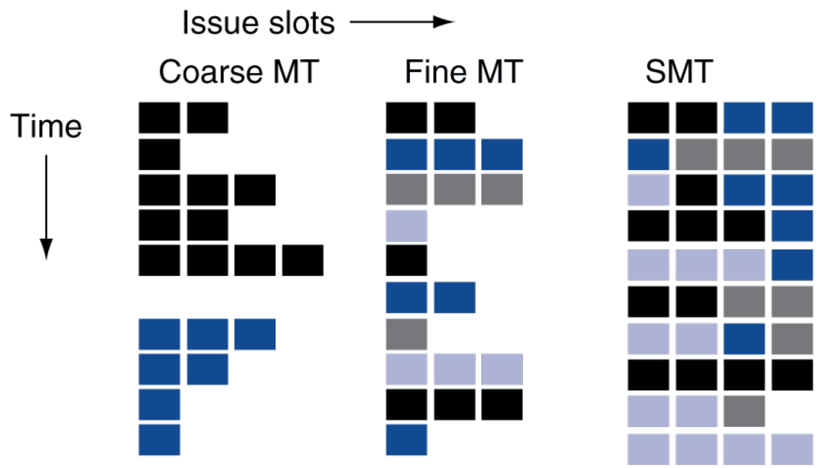


4-way superscalar

Multithreading Example



4-way superscalar



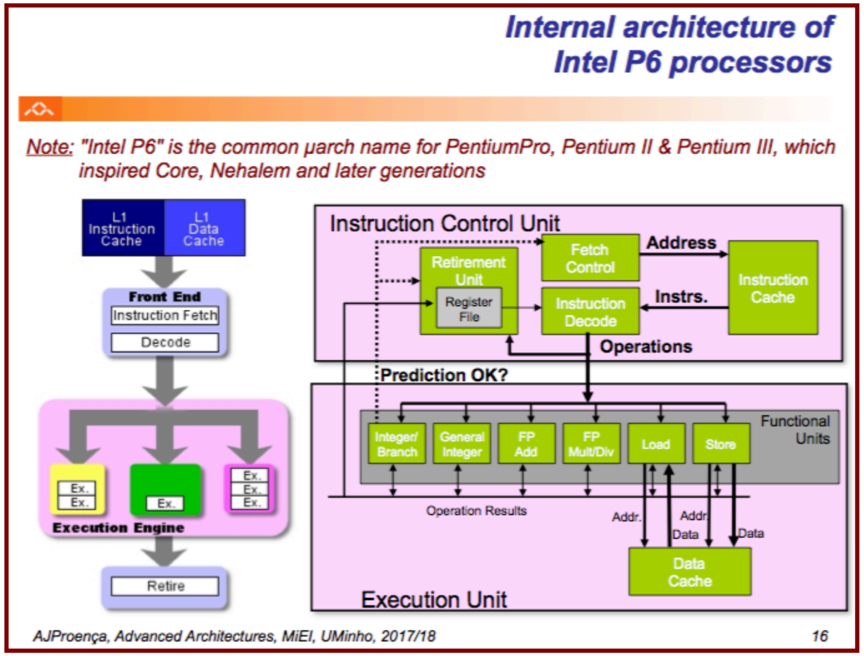
WIKIPEDIA
The Free Encyclopedia

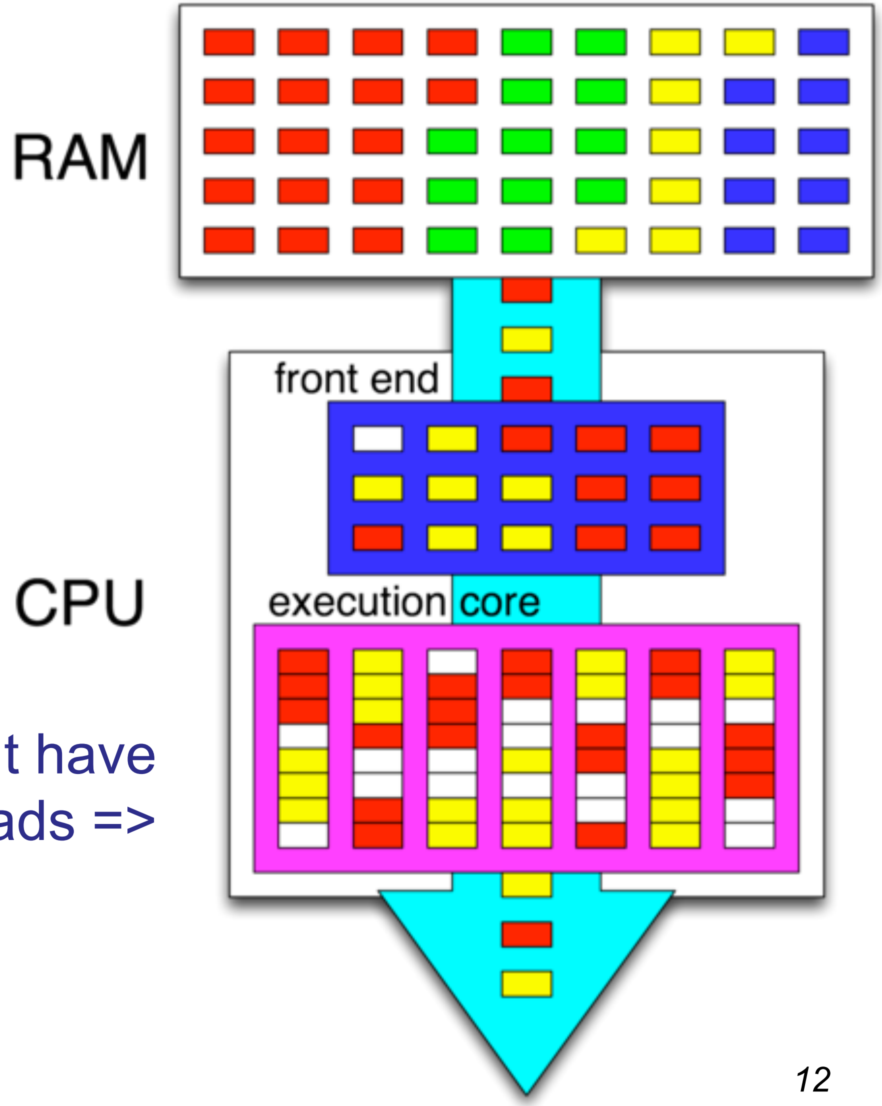
Article Talk

Hyper-threading

From Wikipedia, the free encyclopedia

As seen before...





The pipelined functional units might have better use if shared among more threads =>

Note: white boxes are bubbles...

Reading suggestions (from CAQA 5th Ed)

- Concepts and challenges in ILP: section 3.1
- Pipelining: basic and intermediate concepts: App. C
- Exploiting ILP w/ multiple issue & static scheduling: 3.7
- Exploiting ILP w/ dyn sched, multiple issue & specul: 3.8
- Multithread: exploiting TLP on uniprocessors: 3.12
- Review of memory hierarchy: App. B
- Multiprocessor cache coherence and snooping coherence protocol with example: 5.2
- Basics on directory-based cache coherence: 5.4
- Models of memory consistency: 5.6